

Sorry for missing track of the development steps, here is my roughly development schedule

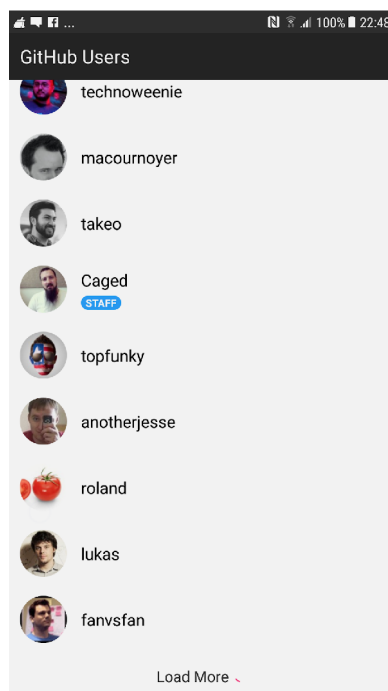
1. 9/28 Implement the architecture of requesting data from GitHub APIs.
2. 9/29 Implement the UI (user list activity and user detail activity) and pagination behavior.
3. 9/30 Error handling (UI for occurring API exception, no network case ...etc.)

Design Consideration:

1. MVP pattern.
  - a. Model: GetUserListUseCase.java, GetUserDetailUseCase.java,  
View: UserListActivity.java, UserDetailInfoActivity.java  
Presenter: UserListViewModel.java, UserDeataillInfoViewModel.java
  - b. Keep the logic of activity as simple as possible.
  - c. Extendibility and testability
2. Using the ViewModel and LiveData to request data and to notify UI the data is ready which can drop unnecessary event according to activity life cycle and prevent memory leak.

Assumption:

Using load more behavior to handle pagination. Requesting data for next page when the last item is visible,



Most difficult in this exercise:

Most of my time spend in this exercise is design the architecture and figure out the behavior of GitHub APIs.

Comment:

認識很多GitHub元老.